CSCC63 Assignment 3

Polytime Reductions, Self-Reductions, and $\mathcal{PS}$

Due [illegible]:59pm, August [illegible]

Warning: *For this assignment you may work either alone or in pairs. Your electronic submission of a PDF to Crowdmark affirms that this assignment is your own work and that of your partner, and no one else's, and is also in accordanc*[illegible]*f Toronto Code of Behaviour on Academic Matters, the Code of Student Conduct, an*[illegible]*iding plagiarism in CSCC63. Note that using Google or any other online resource is* [illegible]

1. (10 marks) Consi[illegible]BLOCKS defined as:

   Instance: A [illegible]ger k.

   Question: We're allowed to permute the rows and columns of M as we like, and we want to do so in a way that minimizes the number of single-number rectangular "blocks" needed to describe the matrix. E.g., is we were to write $(a, b, c, d; 137)$, where $1 \leqslant a \leqslant c \leqslant m$ and $1 \leqslant b \leqslant d \leqslant n$, then we would know that every entry $M_{i,j}$ for $a \leqslant i \leqslant c$ and $b \leqslant j \leqslant d$ would have the number 137.

   Our question is, can we permute the rows and columns of M in such a way that we need no more than k blocks to store it?

   As an example, if M is

$$\begin{bmatrix} 1 & 0 & 1 \\ 2 & 2 & 2 \\ 1 & 2 & 1 \end{bmatrix},$$

   then we can permute the second and third row and the first and second columns to get

$$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix}.$$

   We need four blocks to describe the matri we get from this permutation:

$$\begin{bmatrix} \mathbf{0} & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 2 & 2 \end{bmatrix},$$

$$\begin{bmatrix} 0 & \mathbf{1} & \mathbf{1} \\ 2 & \mathbf{1} & \mathbf{1} \\ 2 & 2 & 2 \end{bmatrix},$$

$$\begin{bmatrix} 0 & 1 & 1 \\ 2 & 1 & 1 \\ \mathbf{2} & \mathbf{2} & \mathbf{2} \end{bmatrix},$$

   and

$$\begin{bmatrix} 0 & 1 & 1 \\ \mathbf{2} & 1 & 1 \\ \mathbf{2} & 2 & 2 \end{bmatrix}.$$

   Notice that we allow the third and fourth blocks to overlap, since they store the same number.

   You will also find that we can't use fewer than four blocks to store this matrix, regardless of the permutation. So if I give you M and k for $k \geqslant 4$, it will be *yes*-instance. If I give you a $k < 4$ it will be a *no*-instance.

You can assume this language is in $\mathcal{NP}$, but you have to show that it is $\mathcal{NP}$-complete (so you have to show that it's $\mathcal{NP}$-hard).

With this in mind, give a reduction that shows [illegible] is $\mathcal{NP}$-hard. Reduce from some language in the Polytime_Reduction_Examples_1 or the Polytime_Reduction_Examples_2 handout.

2. (10 marks) Consider the language CLOSE-SOLUTION-3SAT defined as:

   Instance: A [illegible]ng truth assignment τ for ϕ.

   Question: I[illegible]tisfying truth assignment $\tau' \neq \tau$ such that τ and τ' agree on at least 7/8 o[illegible]nents?

   You can assume t[illegible] but you have to show that it is $\mathcal{NP}$-complete (so you have to show that it's [illegible] in mind, give a proof that CLOSE-SOLUTION-3SAT is $\mathcal{NP}$-hard.

3. Consider the language PATH-CONSTRAINT defined as:

   Instance: A directed graph $G = (V, E)$, two vertices s and t, and two numbers r and k.

   Question: Can we find a size r subset R of vertices in V such that if P_R is the longest simple path from s to t using only vertices in R, then the number of vertices in P_R is somewhere between 2 and $k - 1$?

   (a) (10 marks) Show that PATH-CONSTRAINT is $\mathcal{NP}$-hard (and therefore not likely to be in co-$\mathcal{NP}$).

   (b) (10 marks) Show that PATH-CONSTRAINT is co-$\mathcal{NP}$-hard (and therefore not likely to be in $\mathcal{NP}$). Reduce from some language in the Polytime_Reduction_Examples_1 handout.

   (c) (10 marks) Show that PATH-CONSTRAINT $\in \mathcal{PS}$.

4. (10 marks) Let f be the polytime reduction from TQBF to GG (generalized geography) as described in class.

   Let $\phi = \forall x_1, x_2 \exists x_3, \forall x_4 ((\neg x_1 \vee \neg x_2) \vee x_3 \vee \neg(\neg x_4 \wedge (x_3 \vee x_4))) \wedge x_3 \wedge (\neg x_1 \vee x_4)$.

   Draw $f(\phi)$ (do not simplify the formula before applying f). Label your graph appropriately and indicate the starting node.

5. (10 marks) Recall the STORAGE-BOX language, as given in assignment 2.

   You've already given a proof that this language is $\mathcal{NP}$-complete, but we can also ask how, given an oracle for STORAGE-BOX, to find a solution for a particular STORAGE-BOX instance.

   Show how you can use an oracle for STORAGE-BOX to build a polytime oracle program **FIND-STORAGE-BOXES** such that for any instance of STORAGE-BOX:

   - If any solution exists for that instance, **FIND-STORAGE-BOXES** will return some such solution.
   - If no such solution exists, **FIND-STORAGE-BOXES** will return *null*.

   Justify why your program is correct and runs in polytime.

6. (10 marks) Recall the FEEDBACK-ARC-SET language, as given in the Polytime_Reduction_Examples_1 on the course website.

   You've already been given a proof that this language is $\mathcal{NP}$-complete, but we can also ask how, given an oracle for FEEDBACK-ARC-SET, to find an optimal feedback arc set for a particular directed graph G.

Show how you can use an oracle for FEEDBACK-ARC-SET to build a polytime oracle program **MINIMIZE-FEEDBACK-ARC** such that takes as input any directed graph G and returns a smallest feedback arc set for G.

Justify why your program is correct and runs in polytime.

7. (10 marks) Consider the 3DM language, as given in the Polytime_Reduction_Examples_1 handout on the course website.

   Modify the $\mathcal{NP}$-h to make it a parsimonious reduction from #3SAT.

8. (10 marks) Consid CK-ARC-SET language, as given in the Polytime_Reduction_Examples_1 handout on the co

   Modify the $\mathcal{NP}$-h DBACK-ARC-SET to make it a parsimonious reduction.

9. **Bonus** (10 marks ounded to the nearest multiple of 2.5)

   Consider the SPIRAL-GALAXIES reduction described in the Polytime_Reduction_Examples_2 handout.

   > **Note:** *This handout won't be out at the time Assignment 3 is posted, but it will be up soon.*

   In this reduction we encode a number of widgets that represent wires, inputs, *and*-gates, and other circuit components, and we use those components to build a representation of a 3CNF ϕ. But in that reduction we miss one component: we don't show a way to bend the wires we build, and so we're forced to use a workaround in which the circuit wires always face in the same direction and move laterally by shifting.

   > *Technically we'd also need to modify the crossover widget to get this to work as well, but this is the interesting part.*

   To answer this question, find a way to encode a bent wire (using the endcoding from the reduction) into a SPIRAL-GALAXIES game. Carefully explain why your widget works.